

UNIVERSITY OF TECHNOLOGY SYDNEY

Faculty of Science

**Efficient Solution Methods for Just-In-Time
Machine and Shop Scheduling Problems**

by

Mohammad Mahdi Ahmadian

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE

Doctor of Philosophy

Sydney, Australia

March 2022

Certificate of Authorship/Originality

I, Mohammad Mahdi Ahmadian, declare that this thesis is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the Faculty of Science at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Production Note:
Signature removed prior to publication.

Date: March 29, 2022

ABSTRACT

Efficient Solution Methods for Just-In-Time Machine and Shop Scheduling Problems

by

Mohammad Mahdi Ahmadian

The classical machine (i.e. single and parallel machine) and shop scheduling (i.e. flow-shop, job-shop and open-shop) problems are concerned with performing a set of independent jobs on a given set of machines with or without precedence relations. This thesis explores variants of such problems, pertinent to the practice of Just-In-Time (JIT) manufacturing, where each job (operation) has a due date (or due window) and any deviation from it would incur either earliness or tardiness costs. Embracing JIT philosophy by companies (by discouraging late delivery and reducing warehousing and inventory costs), and their dire need for developing more realistic scheduling models have led to a growing body of research on earliness-tardiness minimization since the late 1970s. Yet, most studies have been devoted to single machine scheduling problems, and very little research has been conducted to address the multiple-machine or shop scheduling settings. Moreover, the current solution methodologies often fail to deliver quality solutions for these problems particularly as the size of instances grows. Therefore, this PhD thesis will contribute to developing efficient algorithms that are capable of obtaining high quality solutions for computationally challenging instances. In addition, we contribute to the existing approaches by integrating exact and heuristic algorithms to maximize the benefits associated with them.

Dissertation directed by Dr. Amir Salehipour

School of Mathematical and Physical Sciences

Dedication

This thesis is dedicated to my mum Mehri Berangi who taught me to never give up.

Acknowledgements

I am extremely grateful to my supervisor, Dr. Amir Salehipour, for his invaluable advice and support during my PhD study. Without his guidance and insightful feedback this thesis would not have been possible.

I would like to extend my thanks to my co-supervisors Prof. Murray Elder and Dr. Leila Moslemi Naeni for their guidance throughout my PhD research studies and their constructive comments on earlier versions of this thesis.

Finally, I would like to thank my parents. Words cannot do justice to express my gratitude for the love and encouragement I have received from them.

Mohammad Mahdi Ahmadian
Sydney, Australia, 2022.

List of Publications

Journal Papers

- J-1. **M. M. Ahmadian**, A. Salehipour and TCE. Cheng, “A meta-heuristic to solve the just-in-time job-shop scheduling problem,” *European Journal of Operational Research*, 2020.
- J-2. **M. M. Ahmadian** and A. Salehipour, “The just-in-time job-shop scheduling problem with distinct due-dates for operations,” *Journal of Heuristics*, pp. 1-30, 2020.
- J-3. **M. M. Ahmadian** and A. Salehipour, “Heuristics for flights arrival scheduling at airports.” *International Transactions in Operational Research* 2020.

Conference Papers

- C-1. **M. M. Ahmadian** and A. Salehipour, “A Matheuristic for Practical Flights Arrival and Departure Scheduling.” *IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pp. 1162-1166. IEEE, 2020.

Contents

Certificate	ii
Abstract	iii
Dedication	iv
Acknowledgments	v
List of Publications	vi
List of Figures	x
Abbreviation	xiv
1 Introduction	1
1.1 Background	1
1.2 Research objectives	2
1.3 Problems addressed	5
1.4 Thesis organization	6
2 Literature Survey	8
2.1 Introduction	8
2.2 Problem statement	9
2.3 Single and parallel machine scheduling	10
2.4 Flow-shop	16
2.4.1 Permutation flow-shop	17
2.4.2 Hybrid flow-shop	19

2.5	Job-shop	21
2.6	Open-shop	28
2.7	Conclusion	29
3	Just-In-Time Single and Parallel Machine Scheduling	31
3.1	Introduction	31
3.2	Problem statement	33
3.3	Relax_1 for ALP	36
3.3.1	Generating an initial sequence	37
3.3.2	Relaxing sequencing constraints for a subset of aircraft	39
3.3.3	Solving partially relaxed sequence	45
3.3.4	Updating the incumbent sequence	51
3.3.5	Operation of the R&S algorithm	51
3.3.6	Computational results	52
3.4	Relax_2 for ASP	60
3.5	Conclusion	63
4	Just-In-Time Job Shop Scheduling	64
4.1	Introduction	64
4.2	Applications	66
4.3	Problem statement	67
4.4	Math_1 for JIT-JSS	70
4.4.1	Sequence encoding and decoding	72
4.4.2	Generating an initial sequence	73
4.4.3	The improvement algorithm	75
4.4.4	Relaxation neighborhoods	76

4.4.5	Re-encoding scheme	79
4.4.6	Computational results	84
4.5	Math_2 for JIT-JSS	88
4.5.1	Improvement algorithm	90
4.5.2	Computational results	94
4.6	Conclusion	110
5	Conclusion	111
5.1	Summary of contributions	111
5.2	Limitations of the study	113
5.3	Future research directions	113
A Comparison of Relax_1 and Relax_2 for ALP and ASP115		
B	Comparison of Math_1 and Math_2 for JIT-JSS	118
	References	120

List of Figures

- 3.1 Operation of the relax procedure in the instance with 15 aircraft and one runway. The relaxed aircraft are represented by green and yellow (the green vertex shows the relaxation center). Aircraft shown in red are immediate predecessor and successor of the relaxed sub-sequence. The conjunctive arcs specify the aircraft that are subject to only scheduling (their sequence is kept as is) and the disjunctive arcs (shown in dashed) represent the relaxed aircraft that are subject to both sequencing and scheduling. Arcs from vertex 7 and arcs to vertex 14 ensure that the relaxed aircraft will be re-sequenced only within the relaxed sub-sequence and that they are connected to the whole sequence. 43
- 3.2 Operation of the relax procedure in an instance with multiple runways, where the relax sub-sequence includes aircraft 9, 10 and 1 (a pair (i, r) represents aircraft i landing on runway r shown in green and yellow (the green represents the relaxation centre)). Aircraft shown in blue and red land on runway one and two respectively. The conjunctive arcs specify aircraft that keep their sequence and are subject to only scheduling. Disjunctive arcs within the relaxed sub-sequence (shown in dashed) represent the relaxed aircraft, which are subject to runway allocation, re-sequencing and scheduling. 45
- 3.3 Only a few immediate precedence constraints might be binding when scheduling an aircraft. 46

3.4	Generating a feasible schedule for the ALP by problem P2: (a) the case of using one constraint (3.4) per aircraft in problem P2 (the default case), (b) the case of using two constraints (3.4) per aircraft, and (c) the case of using three constraints (3.4) per aircraft.	49
3.5	The operation of speed-up procedure for the single-runway case: (a) a relaxed aircraft shown in green is connected either to neighbor aircraft (shown in yellow) that are located within a certain proximity, or to non-neighbor aircraft (shown in gray) by disjunctive arcs, and (b) due to parameter AR an aircraft is only re-sequenced with its neighbors that are within AR positions from the aircraft. The disjunctive dashed arcs highlight that and the conjunctive arcs are used to highlight the non-neighbor aircraft.	50
3.6	A set of aircraft to be relaxed (shown in green and yellow), and a set of non-relaxed aircraft (shown in gray), both of which include the aircraft in the relaxation radius (RR). If the landing penalties of all those aircraft are equal to zero the solve procedure skips relaxing those aircraft and proceeds to the next sub-sequence.	51
3.7	Detailed operations of Relax_1 algorithm for the ALP.	58
4.1	A feasible schedule (of completing the operations) for 4×3 instance. A pair (i, j) represents execution of job i on machine j	71
4.2	The global process of relaxation neighborhoods.	78
4.3	An example of the relaxation neighborhood for 4×3 instance. A pair (i, j) represents job i on machine j . The relaxed jobs are represented by green and yellow (the green vertex shows the relaxation center). Job shown in red is immediate predecessor of the relaxed sub-sequence.	82
4.4	A feasible schedule for the instance $I_{4 \times 3}$. Pair (i, j) represents job i on machine j	83

4.5	Distribution function to choose the value of parameter RC for N_1 . . .	86
4.6	The flowchart of Math_2 for solving JIT-JSS.	92
4.7	An example of the relax-1 neighbourhood for the instance 4×3 . A pair (i, j) represents job i on machine j . As machine 2 has not been selected, its associated operations in R (shown in grey) are performed in the given order by Π . The thick conjunctive arcs impose the order in which grey operations to be performed on machine 2.	93
4.8	An example of the relax-1 neighbourhood for the instance 4×3 . The conjunctive arcs ending at red nodes guarantee the connectivity between relaxed and non-relaxed operations. A pair (i, j) represents job i on machine j	94
4.9	An example of the remove-insert neighbourhood in the instance 4×3 : (a) before the remove-insert operation and (b) after the remove-insert operation.	94
4.10	Swapping two operations in the instance 4×3 ; (a) before the swap and (b) after the swap.	95
4.11	Distribution function 1 to choose the value of parameter RC from for relax-1 and relax-2 and to guide swap and remove-insert neighbourhoods, i.e., to select a pair of positions for the operations in swap and the removal and insertion positions in remove-insert. . .	100
4.12	Distribution function 2 to choose the value of parameter RC from for relax-1 and relax-2 and to guide swap and remove-insert neighbourhoods.	100
4.13	Distribution function 3 to choose the value of parameter RC from for relax-1 and relax-2 and to guide swap and remove-insert neighbourhoods.	101

4.14 Changes in the objective value z under CPLEX, EA, VNS, and Math_2 for “tight-equal-1-20×10”	106
4.15 Changes in the objective value z under CPLEX, EA, VNS, and Math_2 for “loose-tard-2-20×10”.	106

Abbreviation

ACO - Ant Colony Optimization

B&B - Branch-and-Bound

CP - Constraint Programming

GA - Genetic Algorithm

JIT - Just-In-Time

LP - Linear Programming

MIP - Mixed Integer Programming

OM - Operations Management

PSO - Particle Swarm Optimization

R&S - Relax and Solve

SA - Simulated Annealing

TS - Tabu Search

VNS - Variable Neighborhood Search